

Projeto 04 - Inteligência Computacional

Alisson da Silva Vieira

¹Universidade Tecnológica Federal do Parana (UTFPR)
Via Rosalina Maria Dos Santos, 1233 – 87.301-899 – Campo Mourão – PR – Brasil

alisson.v3@hotmail.com

1. Introdução

Neste projeto, foi disponibilizado pelo professor um conjunto de dados BFL [Diego Bertolini 2022]. Nosso objetivo com eles, é analisar o desempenho entre os escritores *Local Binary Patterns* (LBP) e *Gray Level Co-occurrence Matrix* (GLCM). Essa comparação será feita utilizando o algoritmo de *Machine Learning Knn* (K-Nearest Neighbors) [Italo José 2018].

2. Local Binary Patterns

Nesta etapa, foi usado as seguintes partições: 1 quadrante (1 partição vertical e 1 partição horizontal), 4 quadrantes (2 partições verticais e 2 partições horizontais), 9 quadrantes (3 partições verticais e 3 partições horizontais), 16 quadrantes (4 partições verticais e 4 partições horizontais), 25 quadrantes (5 partições verticais e 5 partições horizontais), 36 quadrantes (6 partições verticais e 6 partições horizontais). A Figura 1 mostra os resultados após usar a técnica LBP, que se mostrou não muito eficiente, visto que os resultados foram muito baixos.

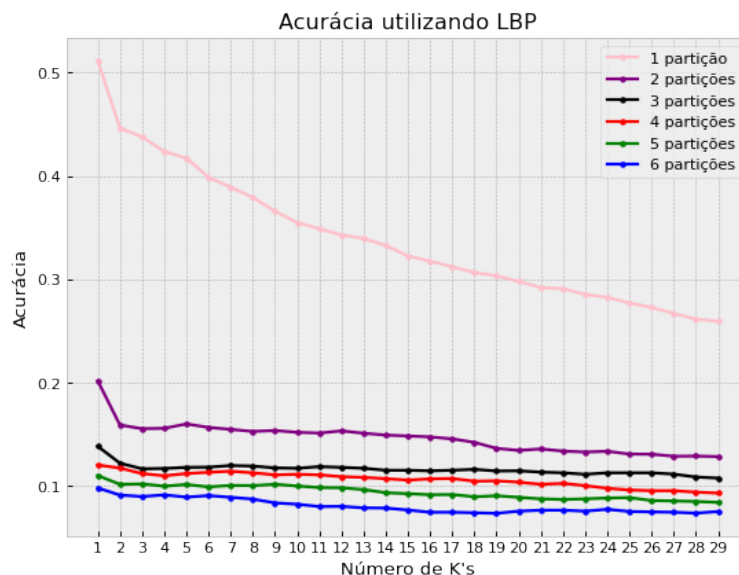


Figure 1. Resultados utilizando LBP.

Pelos resultados, percebemos que a melhor acurácia obtida foi de 0.51% para $k = 1$, utilizando apenas 1 partição. Também se nota uma tendência de queda da acurácia quanto maior for o número de partições, ou seja, para esse problema e para essa técnica,

quanto menor for a partição, melhor são os resultados visto a diferença gritante da queda do desempenho de 1 partição para 2 partições.

3. Gray Level Co-occurrence Matrix

Nesta etapa, foi feito diversos testes para definir a quantidade de quadrantes que maximizaria a acurácia utilizando a técnica, e foi constatado uma diferença nula nas partições testadas: 1 quadrante, 4 quadrantes, 9 quadrantes, 16 quadrantes, 25 quadrantes e 36 quadrantes. Podemos ver o resultado na Figura 2.

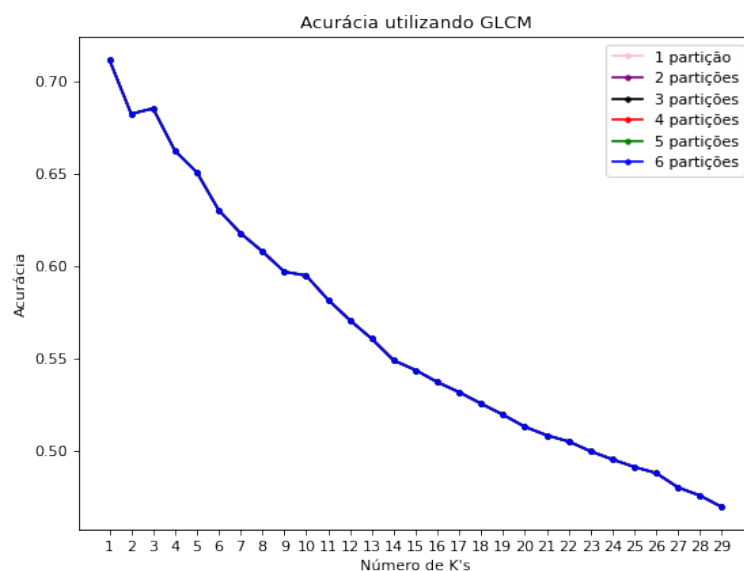


Figure 2. Resultados utilizando GLCM.

Foi constatado que, ao aumentar a quantidade de partições, o custo operacional para realizar o cálculo aumentava consideravelmente. E em relação a acurácia, ela se manteve em 0.71% para $k=1$ independente da quantidade de quadrantes utilizados.

4. Junção

Ao realizar a junção das duas técnicas, esperamos um resultado superior ao já obtido. Foi realizado o particionamento da imagem da mesma forma já relatada, 1, 2, 3, 4, 5 e 6 partições, os resultados são mostrados na Figura 3.

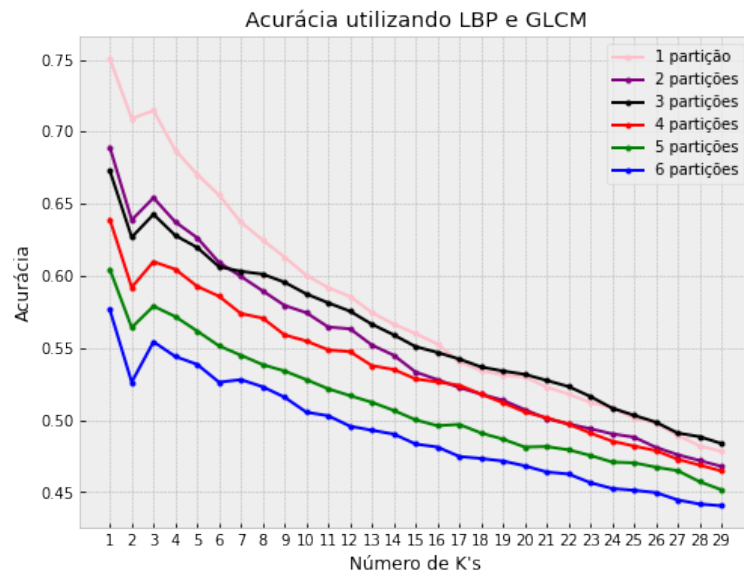


Figure 3. Resultados utilizando LBP e GLCM.

Vemos que o melhor resultado obtido foi de 0.75%, para $k = 1$ utilizando apenas 1 partição. Os resultados obtidos foram os maiores, ultrapassando os 0.71% obtidos ao utilizar o GLCM sozinho.

5. Conclusão

Desta forma, através dessa atividade, conseguimos notar que as vantagens de se utilizar técnicas de *Template Matching* para se obter características das imagens, em conjunto com o algoritmo de aprendizagem supervisionada, Knn. Essas técnicas de *Template Matching* são relativamente simples de se implementar e não da área de *Machine Learning*, e sim da área de processamento de imagens, dessa forma, seu custo depende apenas do processamento das imagens. Pelos resultados, notamos que algumas dessas técnicas possuem bons resultados para esse problema, entretanto, outras não possuem um resultado tão bom. Também podemos dizer que, para esse problema, quanto maior for as partições, menor será a acurácia obtida. Podemos citar como desvantagem de *Template Matching*, o fato de que quanto maior for o tamanho das instâncias, maior será o custo de processamento. Vemos isso na prática, quanto maior for o número de quadrantes maior será o custo para processar as imagens.

References

- Diego Bertolini (2022). Bfl. <https://www.dropbox.com/s/pnctb4cnbbidvok/Base%20BFL%20-%20256x256.zip?dl=0>. Accessed: 3 out. 2022.
- Italo José (2018). Knn (k-nearest neighbors). <https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e>. Accessed: 21 set. 2022.